Paper for the 2002 Command & Control Research & Technology Symposium

Title: *Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems*

Submitted by: Sam Chamberlain
U.S. Army Research Laboratory
ATTN: AMSRL-CI-CT
APG, MD 21005-5067
410-278-8948
Fax: 2934 / 9223
wildman@arl.army.mil
http://www.arl.army.mil/~wildman

Potential Track: Coalition Interoperability

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2002		**00-00-2002 to 00-00-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Army Research Laboratory,ATTN: AMSRL-CI-CT,Aberdeen Proving Ground,MD,21005**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**14**	

Implementation of an Enterprise Identifier Seed Server for Joint and Coalition Systems

Sam Chamberlain

U.S. Army Research Laboratory
ATTN: AMSRL-CI-CT
Aberdeen Proving Ground, MD 21005-5066
410-278-8948
wildman@arl.army.mil
http://www.arl.army.mil/~wildman

Abstract

In any information system, a critical feature is the ability to link together disparate pieces of data and information via relationships. One way to greatly facilitate this task is to provide a common technique for identifying the pieces so that they can be conveniently referenced. Arbitrary linking of data can be accomplished by standardizing one field across disparate data sources. This is the objective of enterprise identifiers (EID). If data can be globally identified using a common scheme, then one can spontaneously reference and retrieve arbitrary pieces of information with minimal prior coordination. A strategy for accomplishing this task was described by the author in a paper [1] at last year's *6th International Command & Control Research & Technology Symposium*. This paper describes the implementation of an EID seed server and some of the ongoing issues encountered and being addressed.

1. Introduction

The term enterprise identifier is the general form of the term enterprise key (EK) that comes from the relational database community. An EK is a surrogate key[1] (SK) that is unique across the whole enterprise; in other words, it is a universal SK (USK). For example, if the enterprise is the DoD, then no two data items may have the same EID within the DoD unless they represent the exact same object. To facilitate interoperability, the format of an EID must be technology independent. It must be usable by relational databases, object databases, formatted flat files, or any other formatted source of data. This is an important characteristic for integrating disparate data systems. For a list of sources of information on topics related to implementing EIDs, see [2].

When a database management system creates (e.g., inserts) data, it must obtain an EID. This EID is attached to the data for its lifetime, that is, it is a persistent label. Because the data has an EID, no matter where it propagates within the enterprise, its EID is guaranteed never to collide with any other EIDs already resident in any database[2]. EIDs serve the same purpose in data sources that

---

[1] Surrogate Key: a single attribute, candidate key from which no information can be gleaned about the entity it identifies.

[2] The phrase "within the enterprise" refers to the other data sources that are also using the same EID scheme.

Uniform Resource Identifiers (URI) serve in the web community[3]. However, this scheme for EIDs maintains three important characteristics:

- All EIDs are surrogate keys (fit in a single attribute, no embedded intelligence),
- All EIDs have the same size and form (in this implementation 64-bits),
- The size is chosen to be as small as possible.

The reason for these characteristics is to support transactions over a wide variety of military data systems that may have to operate in constrained communication environments. Low bandwidth or degraded communication conditions will likely exist in wireless, military systems for a significant duration. Any URI scheme adopted for military use must also operate under these conditions. With a few simple modifications, this EID scheme can be easily modified to meet registration requirements as a URI, but at the expense of substantial increases in bandwidth and space utilization.

The primary challenge in creating an EID system is developing an approach to guarantee that EIDs are unique, while ensuring that they can be easily obtained and cause no performance degradation. An additional objective is to provide a flexible scheme for allocating EIDs that allows the system managers as much control and freedom as possible. A common approach for producing globally unique values is by concatenating two smaller values. In this implementation, devices called EID Servers (ES) create globally[4] unique EIDs by appending a locally managed suffix to a globally unique prefix obtained from an authoritative source. The prefix is called an *EID Seed* and the authoritative source is called an *EID Seed Server* (*ESS*). Figure 1 illustrates this approach.

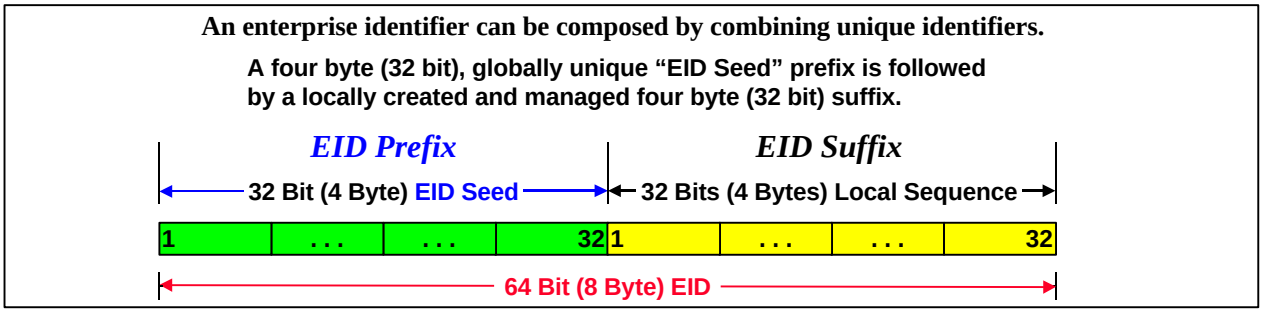

Figure 1: EID Composition

Figure 2 illustrates the EID allocation architecture. It includes three levels of components: EID users (on the right), EID servers (in the middle), and EID seed servers (on the left). EID users (usually databases) "connect" to an ES to obtain EIDs. An ES is *any* computer program that provides EIDs to requestors. It can be 20 lines of Java™ code (as one might find embedded in a PDA[5]) or a complex program that resides on a separate machine and serves many databases. The topic of this paper is the ESS that passes out 32-bit EID seeds to its registered users. This allows

---

[3] See: http://www.ietf.org/rfc/rfc2396.txt for the latest RFC on URIs and http://www-old.ics.uci.edu/pub/ietf/uri/ for a comprehensive list of related topics.

[4] The term "globally" refers to all the members of the enterprise.

[5] PDA – Personal Digital Assistant, for example, a Palm Pilot.®

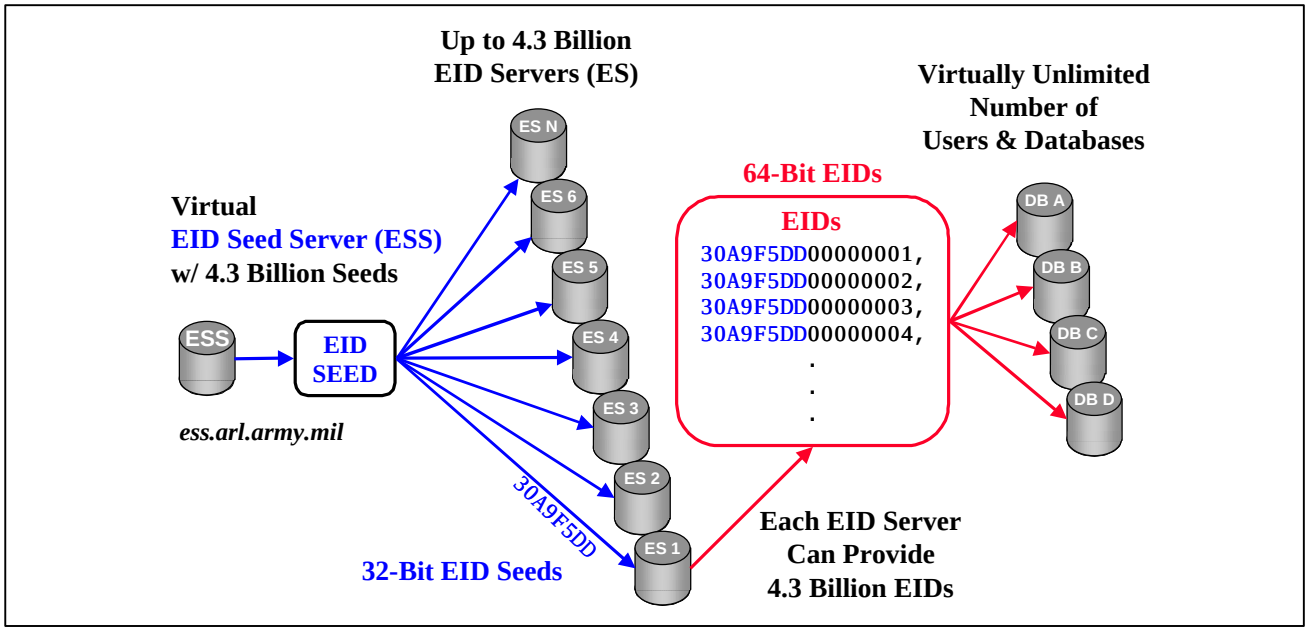

Figure 2: An EID Server Architecture

nearly 4.3 billion EID servers to be established, each able to dispense 4.3 billion EIDs for a total allocation of 18 billion-billion EIDs.

To avoid confusion, one must understand the difference between EIDs, EID seeds, EID servers, and EID seed servers. To reiterate these differences, an EID is a 64-bit value that is used to uniquely identify a database record or object. Internally, it may be represented as a 64-bit binary value, two 32-bit binary values[6], or as sixteen hexadecimal characters.[7] EIDs are obtained from EID servers (ES) that create them by appending a locally controlled 32-bit suffix to a 32-bit prefix. The prefix is an EID seed that is obtained via an EID seed server (ESS) on which one must have an account. Observe that the reason that an ES can produce nearly 4.3 billion EIDs is because of the size of the locally controlled suffix. The fact that the prefix and suffix are both 32-bits is a coincidence, or perhaps better stated, they are convenient numbers to use with computers. Any sized prefix and suffix will work. The reason 64-bits was selected was that it was determined that this was the smallest value that could accomplish the task. Larger values will work equally well, but due to bandwidth considerations, the smallest possible values were selected.

2. Accessing the EID Seed Server

In April 2002, the US Army Research Laboratory (USARL) established a prototype ESS. To obtain EID seeds, one must obtain an ESS account. Obtaining an account is simple, is open to anyone who wishes to apply, and is accomplished by visiting the URL https://ess.arl.army.mil/. When one applies, they must first agree to abide by four basic criteria — they will ensure that:

---

[6] Some databases do not yet support 64-bit binary values.

[7] In hexadecimal notation, an ASCII character represents a pattern of four bits. Therefore, the 16 ASCII characters 0-9 and A-F are used to denote the 16 combinations of four bits: 0000 through 1111. Note that it is a coincidence that 16 characters are also required to represent a 64-bit EID.

1. their point of contact information is always kept current,
2. the ESs they establish produce 64-bit EIDs using a bona-fide EID seed obtained from an ESS as a prefix,
3. EIDs are maintained in either binary form as a single 64-bit field, or two 32-bit fields, or as 16 characters using hexadecimal notation, and
4. the ES includes a mechanism to guarantee that EIDs are never duplicated. This implies that the ES must have some type of backup scheme to prevent re-use in the event of a power loss or major malfunction.

Once one has agreed to these four conditions, the requester is transferred to a registration page to enter the usual official contact information. The current page is illustrated in Figure 3. Each user recommends a unique user name to allow them to log on to the ESS. All contact information can be updated; but two items, organization and email address, require the concurrence of an ESS account administrator. The reason for this is to maintain control of who actually is using the ESS. If organizational email addresses are used, the account representative may change for an organization without any ESS administrator involvement. The "Region" and "Province" fields are to facilitate international users. New fields can be easily added as required.

Once the request is submitted, an email is sent to the email address specified that includes information on how to validate the email address. This uses the common validation technique of having the requestor log on to a machine to enter a code sent in the email. Once this is accomplished, an email is sent to the ESS administrators notifying them that an account request is pending from a validated source. An ESS administrator may then approve the ESS user account.

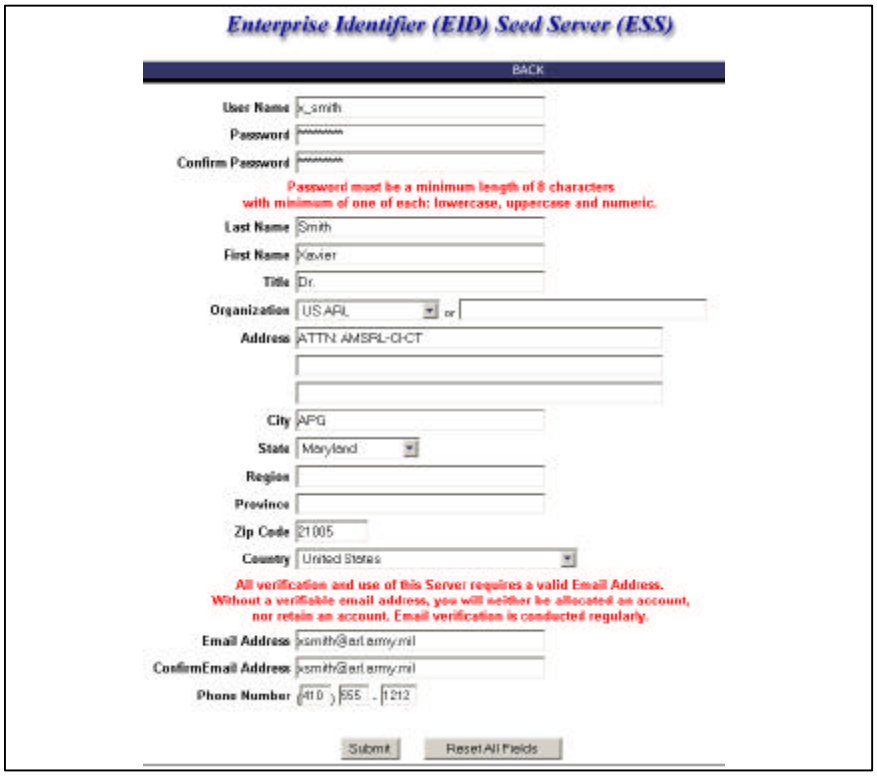

Figure 3: The Current ESS Registration Page

Anyone from any organization or country may request an account. Because a goal of the prototype is to encourage the use of EIDs and to gain an experience base, it is unlikely that anyone will be turned down. Eventually, this may change and accounts may be scrutinized to limit subscribers to those with a bona fide requirement or official capacity to create official data. Some association with the US Department of Defense (DoD) could be required (e.g., one is a member of, or doing work for, a defense establishment). Examples are officials in corporate information offices (CIO); acquisition organizations or companies, such as program executive officers (PEO), project managers (PM), and systems builders; and the simulation community. To handle non-traditional cases, a human will control the account authorization process. This is the role of the ESS account administrators. The current requirement is to respond to a request within 24 hours, although the goal is to respond within minutes.

Once approved, a requestor becomes a subscriber and may request EID seed accounts. For most people, one account is sufficient. However, it is envisioned that a single person may require several accounts because they are handling several end-users or may have security considerations. Consequently, a user may set up several accounts. To create an EID account, a subscriber logs on to the ESS and goes to the "New Account" page. This page is illustrated in Figure 4. An EID seed account does not have to have a globally unique name since it is associated with an ESS user account. Therefore, ESS users can select any name that is meaningful to them.

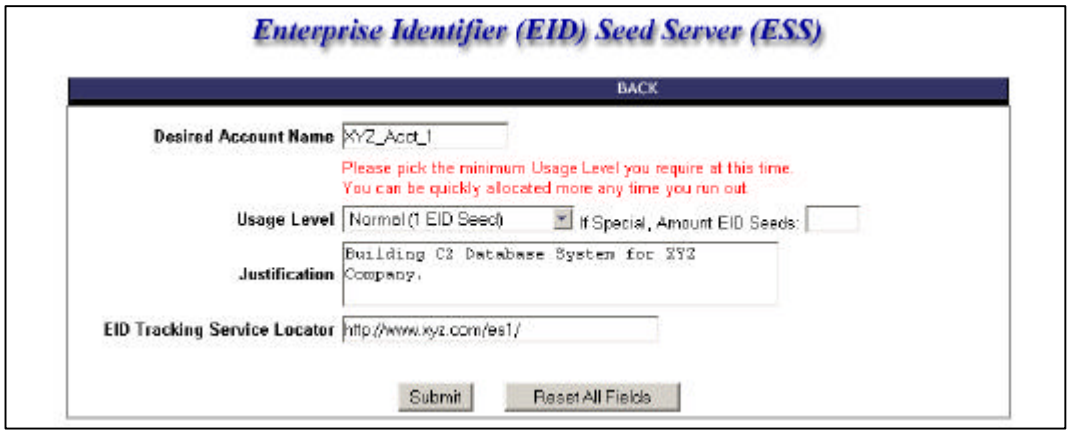

Figure 4: The New EID Account Page

2.1 Usage Level

There are four usage levels available for an EID seed account: Normal (1 EID Seed), Moderate (10 EID Seeds), Heavy (100 EID Seeds), and Special (> 100 EID Seeds). The objective is to provide the EID seed user with as many seeds as they require, but no more. For this reason, users are encouraged to be frugal and may increase their usage level at a later time. Recall that one EID seed supports the creation of 4.3 billion EIDs. Therefore, for many users one EID seed is enough. If they use up their 4.3 billion EIDs, they simple increase their EID usage and obtain another EID seed.

As explained in [1], the primary reason for requiring more than one EID is database performance. The use of EIDs must not cause any database performance degradation or system management problems. If a set of database management systems (DBMS) is tightly coupled over a high

bandwidth communications environment, then a single ES (established with a single EID seed) may be adequate to serve many systems without degradation. However, if one has many distributed systems that are widely dispersed, then an EID seed may be required for each system. This is the expected situation for battle command systems and is the environment for which this implementation is based. If a project manager is building 12,000 tactical systems, then 12,000 EID seeds may be required. If every person has a wearable computer, then every one of those systems may require a separate EID seed.

Notice, however, that the DBMS dispersion problem can be addressed in another way. If every system is likely to create 4.3 billion pieces of data, then a separate EID seed, or EID prefix, is required for each system. This equates to solving the problem at the ESS level. The alternative is for the system manager to solve the problem at the ES level by allocating the locally controlled, EID suffix into blocks. For example, the 4.3 billion EID suffixes could be broken into 100 43 million EID blocks and dispersed into 100 distributed systems. As a system nears the end of its block of EIDs, the manager obtains a new EID seed and creates another set of blocks to be distributed. It is this flexibility that is afforded by the properties of surrogate keys (i.e., no embedded intelligence) that makes managing EIDs so simple.

Just as with ESS user accounts, an ESS administrator approves and enables individual EID seed accounts. Once approved, the ESS user may obtain EID seeds up to the maximum number authorized for the EID seed account without communicating with an ESS administrator. EID seeds are obtained using the "Get New Seed" page as is illustrated in Figure 5. This page provides information on the usage of the EID seeds for a particular account. In this case, it states that this account has 1 EID seed remaining. The user can request up to that number. When this amount is entered and the "GET EID" button is pressed, the ESS allocates that number of EIDs to the EID seed account. Simultaneously, the user is sent an email stating that EID seeds were allocated to the specified account.

When users log on to the ESS, they are placed in their ESS User Area (i.e., homepage) as is illustrated in Figure 6. This page lists all of their accounts, their status, the number of EID seeds remaining, and the number of EID seeds obtained. A user can acquire a list of their EID seeds by selecting the "List" text as is illustrated in Figure 7. In this example, this account has EID seed "01000002" that was allocated to it on 11 April 2002. Therefore, this ESS user may establish an

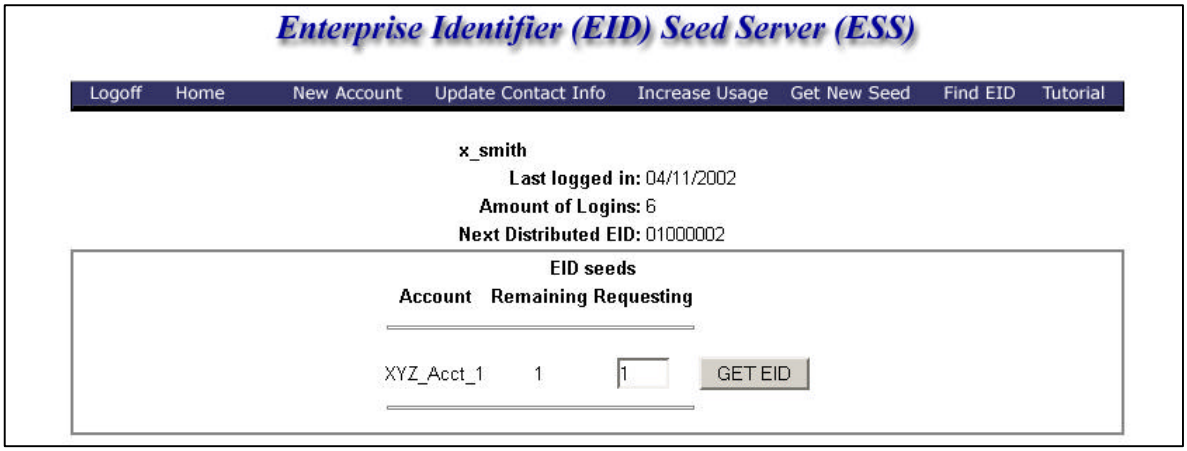

Figure 5: The Get EID Seed Page

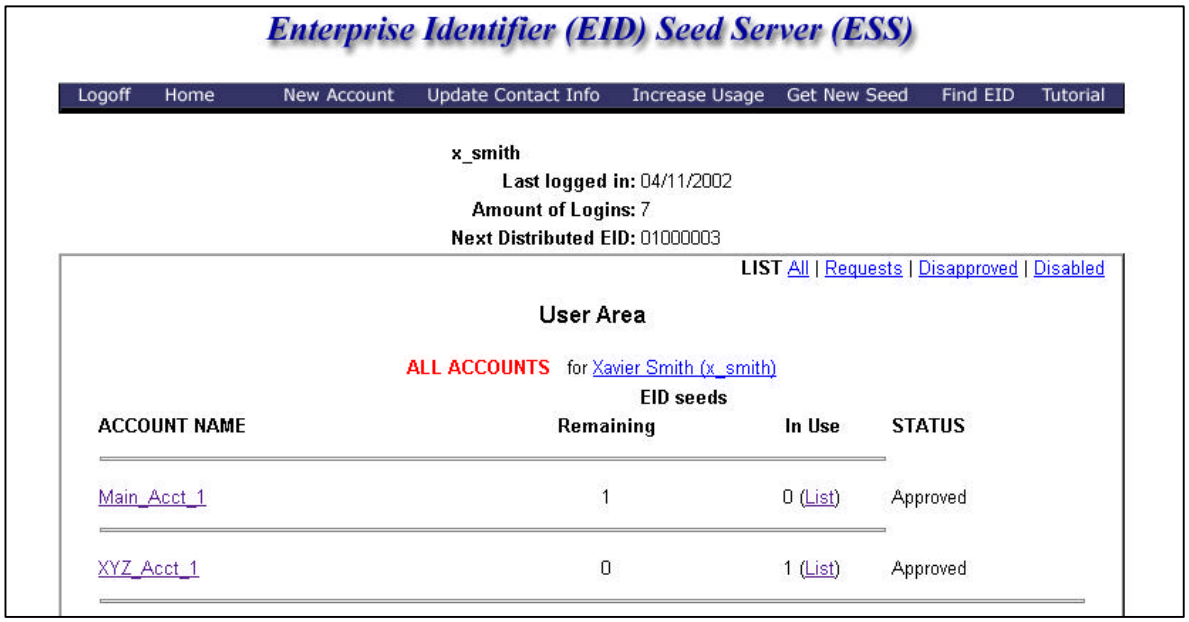

Figure 6: ESS User's Area (Home Page)

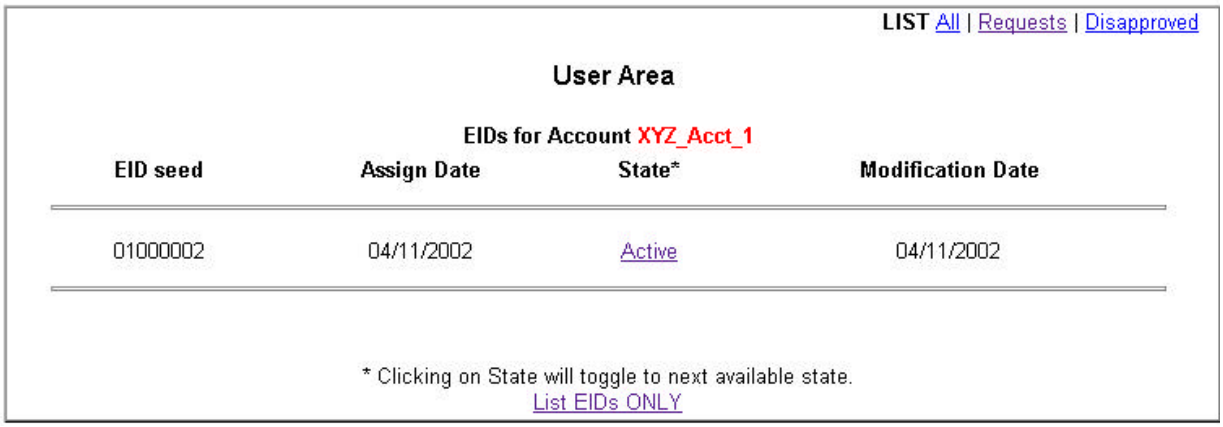

Figure 7: Listing the EID Seeds of an EID Seed Account

ES and assign this seed to be its EID prefix. The ESS guarantees that no other ESS subscriber has been assigned this seed, and therefore, no EID created using it will ever collide with any EID created by another ESS subscriber. The status field and modification date will be described in the following section on EID tracking.

An interesting ramification of this approach is that *the enterprise* is now defined as the set of ESS subscribers. This set is currently not constrained by service, governmental, or national boundaries. This facilitates an open medium by which to experiment with and experience the advantages of EIDs. It is the author's experience that standardization edicts are rarely successful. Consequently, the use of EIDs is proposed as a strictly voluntary action. However, it is expected that EID users will rapidly realize the advantages of using EIDs and the practice will quickly spread. For two organizations to implement EIDs, all that is required is an agreement by the parties to get ESS accounts and abide by the four basic rules. Further, as explained in [1], although there are significant advantages for database maintenance when using EIDs as primary keys, all the

interoperability advantages are obtained when they are used as alternate keys. Thus, EIDs can be implemented without modifying the existing primary key system in a legacy database. If the legacy system is a relational database, all that is required is the addition of a single new column in each data table to contain an EID. It is then a simple procedure to assign an EID to each row.

2.2 EID Tracking

With any distributed DBMS, procedures should be implemented to ensure that references to data are accompanied by the data referenced. In other words, if a field in one database table references a row of another database table, then that row must be present. If this is violated, a referential integrity error occurs. This situation is no different for EIDs as other types of keys. Although it should rarely happen, it is possible to be given an EID that cannot be found in the distributed database. However, since EIDs have no embedded intelligence, it is often impossible to devise a scheme to track down the data item that caused the referential integrity error. To facilitate a solution to this problem, an EID tracking system is being developed.

Although an EID has no embedded intelligence, it is known that an EID prefix is an EID seed that was obtained from the ESS. Therefore, one can at least discover who was given an EID seed by contacting the ESS. Current ESS policy is that the contact information associated with an ESS user account is available to all subscribers. This feature is executed using the "Find EID" option. Figure 8 illustrates both a "Find EID" page and a corresponding response page. In this example, an EID of "0100000200050FFE" is entered, and the corresponding contact information is provided. The EID prefix is parsed from the EID (first 8 hexadecimal characters) and that EID seed is used to query the ESS. In this case, the basic contact information entered by the ESS user is returned: name, phone number, and email address. It would be up to the inquirer to then contact the EID seed owner and ask about the data item tagged with the EID "0100000200050FFE." Of course, it is totally at the EID seed owner's discretion to divulge the information. This is similar to IP addresses. One is not *required* to register an IP address with a hostname via a domain name

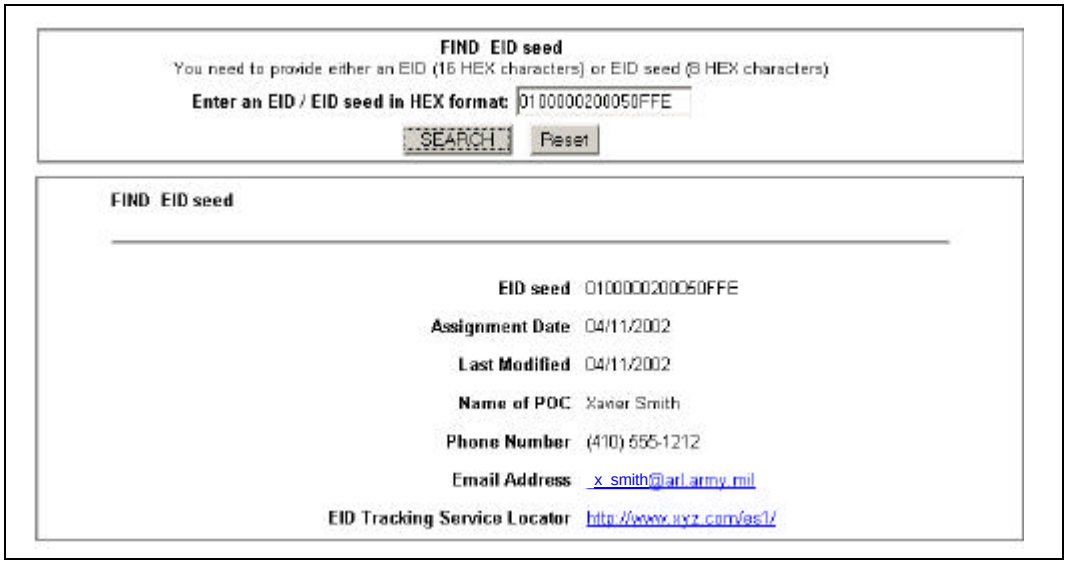

Figure 8: The Find EID Request and Response Pages

service. However, most people do this to allow their hostname, or URL[8], to be used instead of a raw IP address. The same practice is true with EIDs.

What is desirable is to have an automated system to do EID searches. This is the function of the EID Tracking Service Locator (TSL). In Figure 8, a URL has been entered to indicate that this ESS user has implemented a tracking system for some or all of the EID seeds it has obtained. This allows the system to automatically forward the "Find EID" command to the URL (along with the source information about who is asking). This capability, and the associated protocol specification, is currently under development. The concept is that this process continues until the DBMS containing the data is discovered. Then, a "Get EID" command is issued to the DBMS and the resulting data is sent to the requestor in the form of an XML document[9]. This requires that a mechanism be developed to convert the "Find EID" command into a local "Get EID" command that executes a query in the native database query language of the appropriate DBMS and converts the result into XML code to be returned to the requestor. In other words, a tracking server system must include a "Get EID" function that interacts with the local DBMS's.

A recursive EID tracking system is illustrated in Figure 9. This example shows several different levels of complexity that are based upon the implementation decisions of the system manager. The line marked with an 'A' shows the simplest level that occurs when an ES (labeled "ES3") is

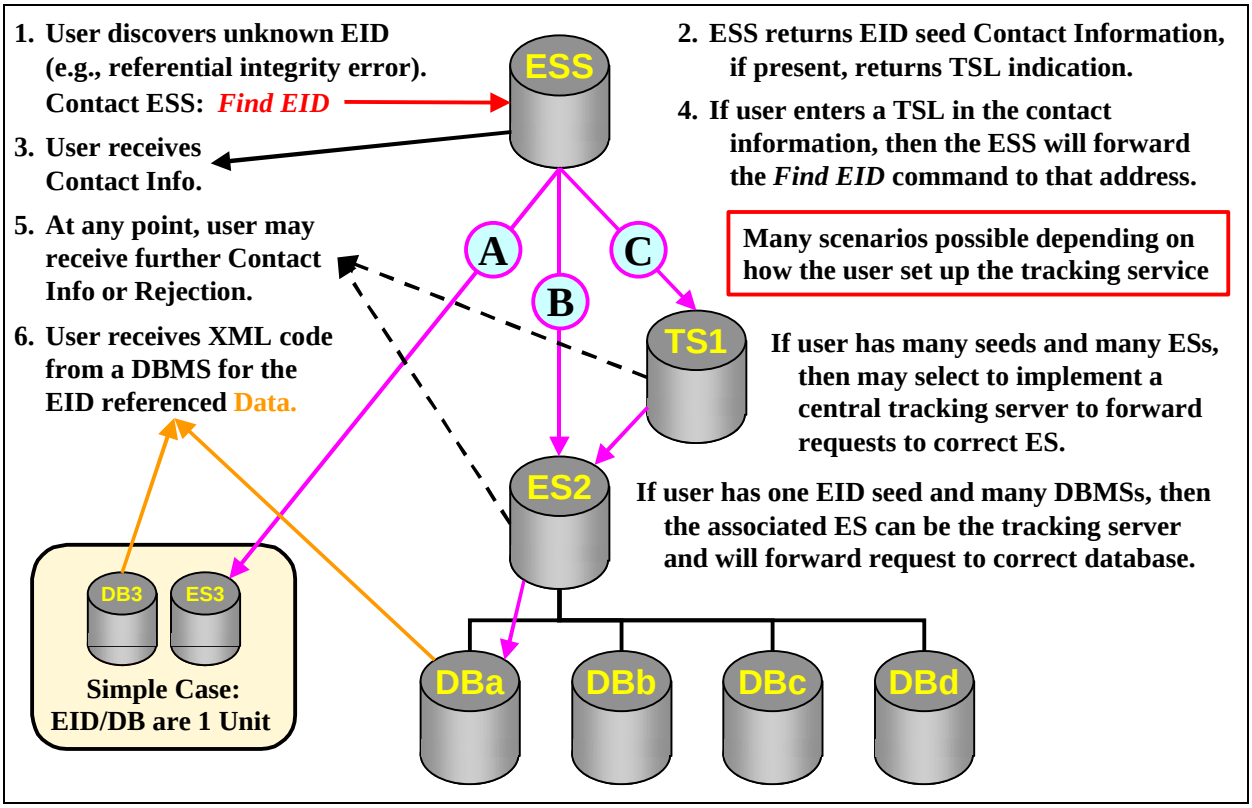

Figure 9: A Recursive EID Tracking System

---

[8] URL: Uniform Resource Locator, more commonly known as a web address.

embedded in a DBMS. The user's TSL allows the ESS to redirect the "Find EID" request directly to the source of the data. If the user is tracking the listed EID, then the ES executes a Get EID command to the database, which in turn generates the XML code that is returned to the requestor.[10] In actuality, this is probably the most likely scenario because the single EID seed with a single independent database is expected to be the most common situation. Under this condition, the data that is tagged with the requested EID is discovered with only a single redirection by the ESS of the Find EID operation.

The line marked with a 'B' illustrates a slightly more complex level in which the ES manager has decided to implement EID suffix blocking as was explained in Section 2.1. Because only the user (not the ESS) has any knowledge of the blocking (or any allocation) scheme, a "Find EID" request must first be directed to the ES for further redirection. In this case, the ES (labeled "ES2") forwards the "Find EID" command to the correct DBMS (e.g., DBa). Alternatively, it could forward a "Get EID" command to the appropriate DBMS and handle the results of the query by converting it into XML code and returning it to the requestor.

The line marked with a 'C' illustrates the most complex level. In this case, a user has an EID seed account with multiple EID seeds each with its own ES. When this occurs, the user must have an intermediate tracking service established to forward the "Find EID" command to the correct ES. This is because of the decision to associate the TSL with an EID seed account, and not with the individual EID seeds. It does not matter if a user has a normal, moderate, heavy, or special EID seed account, each account may be associated with only a single TSL. Originally, it was considered to allow a TSL to be designated for each EID seed, but it was decided that this would place an unnecessary burden on the ESS and also on the ESS users who would have to contact the ESS for what are essentially local changes to their tracking service. By having a single TSL per EID seed account; the burden of effort is divided equitably between the ESS and the user's local systems.

Observe that from the ESS's perspective, all three of these cases are the same. When a "Find EID" command is executed, the ESS simply retrieves the EID seed account under which the EID was allocated. If a TSL field entry exists, it forwards the "Find EID" command to the URL entered in the TSL field. In reality, all these examples are figurative since there are numerous ways to implement tracking servers and ESs. All functions may reside on a single machine (with a single hostname), each on a separate machine, or some combination of machines. It is this flexibility that allows users to manage their data and the allocation of EIDs as they determine locally. This allow the ESS to be as unintrusive as possible.

An ESS characteristic that was debated is whether visibility of user contact data should be mandatory (as it is currently), or whether it should be user selectable. In other words, should the ESS users be able to specify whether the "Find EID" command returns their contact information. One argument is that there may be users who do not wish to be identified (e.g., intelligence organizations). It was decided that the ESS should remain completely open and independent of any

---

[9] XML: The Extensible Mark-up Language, a standard for describing the structure of information; see http://www.w3.org/XML/.

[10] Alternatively, the DBMS could intercept the Find EID command and execute this process.

organizational security issues (another advantage of the properties of surrogate keys). If an ESS user does not want to be identified, then a proxy can always be hired to be their ESS account manager (one of the reasons for allowing multiple EID seed accounts per user). Having a "front" is not a novel idea in many businesses. Many organizations specialize in the practice of masking the identity or true character of other organizations. This is not the business of the ESS. All that is required is that its users abide by its four stipulations, which includes maintaining current contact information.

Observe that there are no guarantees that a tracking service will be provided for an EID. All that is guaranteed is that contact information for an EID seed will be provided. A user is not required to provide a TSL or any tracking services. Furthermore, a user can arbitrarily provide tracking for some EIDs and not others. This means that a requestor may receive a refusal notice at any level of the tracking function exemplified in Figure 9. Note that this is also true when tracking IP addresses within the Internet.

Finally, there is a user settable parameter to indicate the activation state of an allocated EID seed. This is a toggled value that a user can set for each EID seed as is illustrated in Figure 10. There are several reasons for this field, the most prominent being the time lapse between the acquisition of an EID seed and the establishment of an ES that uses that seed. When a user executes the "Get EID" command (see Figure 5), the EID status is set to "inactive." Once an ES is established and is activated, the user can toggle this field to indicate that the EID seed is now in use. Currently, this parameter is used only as an indicator (e.g., to display the status). So a user can toggle this variable without any effect on the ESS. For example, a user may change the state to "inactive" to indicate

User Area

EIDs for Account XYZ_Big_Acct

EID seed	Assign Date	State*	Modification Date
01000003	04/12/2002	Active	04/12/2002
01000004	04/12/2002	Active	04/12/2002
01000005	04/12/2002	Active	04/12/2002
01000006	04/12/2002	Inactive	NOT MODIFIED
01000007	04/12/2002	Active	04/12/2002
01000008	04/12/2002	Active	04/12/2002
01000009	04/12/2002	Active	04/12/2002
0100000a	04/12/2002	Inactive	NOT MODIFIED
0100000b	04/12/2002	Inactive	NOT MODIFIED
0100000c	04/12/2002	Inactive	NOT MODIFIED
0100000d	04/12/2002	Active	04/12/2002
0100000e	04/12/2002	Active	04/12/2002
0100000f	04/12/2002	Inactive	NOT MODIFIED
01000010	04/12/2002	Inactive	NOT MODIFIED
01000011	04/12/2002	Inactive	NOT MODIFIED
01000012	04/12/2002	Active	04/12/2002

Figure 10: EID Seed Activation

that an ES is no longer active or operational. This does not automatically imply that the tracking services for the ES are switched off, but only that it is no longer available for dispensing EIDs. Eventually, this value may be used in ESS functions like "Find EID." Future uses for this field are still being discussed.

3. Summary

To achieve interoperability across automated information systems (AIS), a critical feature is the ability to link together disparate pieces of data and information. One way to greatly facilitate this task is to provide a common technique for identifying data and information so that they can be conveniently referenced. Arbitrary linking of data can be accomplished by standardizing one field across disparate data sources. This is the objective of enterprise identifiers (EID). If data can be globally identified using a common scheme, then one can spontaneously reference and retrieve arbitrary pieces of information with minimal prior coordination. This can be accomplished using alternate keys, thus having little (or no) affect on legacy data systems.

EIDs serve the military database community in the same way that URIs serve the web community. This implementation of EIDs maintains three important characteristics:

- All EIDs are surrogate keys (fit in a single attribute, no embedded intelligence),
- All EIDs have the same size and form,
- The size is chosen to be as small as possible.

The reason for these characteristics is to support transactions over a wide variety of military data systems that may have to operate in constrained communication environments. Any URI scheme adopted for military use must also operate under these conditions. With a few simple modifications, this EID scheme can be easily modified to meet registration requirements as a URI, but at the expense of substantial increases in bandwidth and space utilization.

This paper describes an implementation of a prototype EID seed server (ESS) and some of the challenges encountered that are being addressed. Primary characteristics include simplicity, flexibility, and unintrusiveness. To accomplish these characteristics, a three-tiered approach is used that is based upon the common idea of producing globally unique values by concatenating a locally managed unique value to an existing globally unique value. Perhaps the most difficult task in implementing EIDs is obtaining an agreement within the enterprise to use them. For this reason, it is recommended that the use of EIDs continue to be voluntary until users have an adequate opportunity to try them and experience their benefit for themselves. In this implementation of EIDs, the enterprise is defined as the set of ESS subscribers. Perhaps no other single agreement between data managers will have a larger payoff than agreeing on this very basic feature of their data.

4. References

[1] S. Chamberlain: "An Enterprise Identifier Strategy for Global Naming Across Arbitrary C4I Systems," Proceedings of the 6th International Command and Control Research and Technology Symposium; US Naval Academy, Annapolis, MD; 19-21 Jun 2001. http://www.dodccrp.org/6thICCRTS/Cd/Tracks/Papers/Track2/059_tr2.pdf

[2] Papers related to Surrogate Keys, Enterprise Keys, and Enterprise Identifiers:

- E.F. Codd:
  - "Extending the Relational Model to Capture More Meaning," *ACM Transactions on Database Systems, Vol 4(4),* Dec 1979.
- C.J. Date:
  - "An Introduction to Database Systems, Volume II," Addison-Wesley, Reading, MA, 1983; ISBN 0–201–14474–3
  - "Relational Databases: Selected Writings," Addison-Wesley, Reading, MA, 1986. ISBN 0–201–14196–5
- T. Johnston:
  - "Primary Key Reengineering Projects: The Problem;" DM Review, February, 2000, http://www.dmreview.com/master.cfm?NavID=55&EdID=1866.
  - "Primary Key Reengineering Projects: The Solution;" *DM Review*, March, 2000, http://www.dmreview.com/master.cfm?NavID=55&EdID=2004
  - "De-Embedding Foreign Keys," *DM Direct* (online): Part 1, June 2, 2000. http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2308.
  - "De-Embedding Foreign Keys," *DM Direct* (online): Part 2, June 9, 2000. http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2322.
  - "De-Embedding Foreign Keys," *DM Direct* (online): Part 3, June 16, 2000. http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2331.
  - "De-Embedding Foreign Keys," *DM Direct* (online): Part 4, June 23, 2000. http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=2341.
- M. Lonigro, Mike:
  - "The Case for the Surrogate Key", Intelligent Enterprise Database Programming and Design On-line, May 1998; http://www.dbpd.com/vault/9805xtra.htm.
- "Enterprise Identifiers For Logistics - An Approach in Support of Army Transformation Initiatives," http://arch-odisc4.army.mil/data_mgt/docs_org_id/ArmyLogisticsStudy_xFinal.pdf